

FIGURE 1

procedure Greedy ($T(X)$, \bar{e} , G , θ)

Input: n -attribute table T and n -vector of error tolerances \bar{e} ;
Bayesian network G on the set of attributes X and
threshold θ on the relative benefit for selecting a
CaRT predictor.

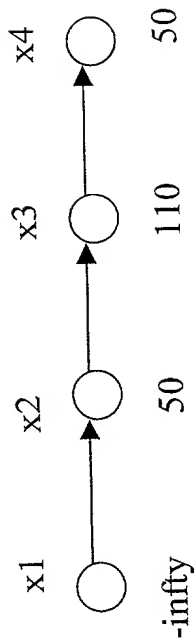
Output: A set of materialized (predicted) attributes X_{mat} (X_{pred}
 $= X - X_{\text{mat}}$) and a CaRT predictor for each $X_i \in X_{\text{pred}}$.

begin

```

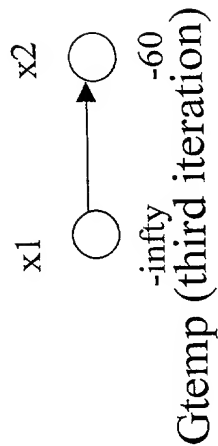
1.   $X_{\text{mat}} := X_{\text{pred}} := \emptyset$ 
2.  let  $\langle X_1, X_2, \dots, X_n \rangle$  be the attributes in  $X$  sorted in
    topological order of  $G$ 
3.  for  $i := 1, \dots, n$ 
4.  if  $\pi(X_i) = \emptyset$  then  $X_{\text{mat}} := X_{\text{mat}} \cup \{X_i\}$  /*  $X_i$  must be
    materialized if it has no parents in  $G$  */
5.  else
6.   $M := \text{BuildCaRT}(X_{\text{mat}} \rightarrow X_i, e_i)$ 
7.  if  $(\text{MaterCost}(X_i) / \text{PredCost}(X_{\text{mat}} \rightarrow X_i)) > \theta$  then  $X_{\text{pred}} :=$ 
     $X_{\text{pred}} \cup \{X_i\}$ 
8.  else  $X_{\text{mat}} := X_{\text{mat}} \cup \{X_i\}$ 
9.  end
10. end
end
```

FIGURE 2: The Greedy CaRT Selection Algorithm



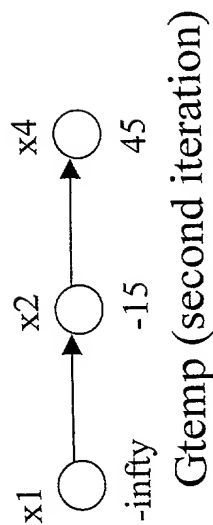
Bayesian Network G

FIGURE 3A



(b) Gtemp (first iteration)

FIGURE 3B



Gtemp (second iteration)



Gtemp (third iteration)

FIGURE 3C

FIGURE 3D

procedure MaxIndependentSet ($T(X)$, \bar{e} , G , neighborhood())

Input: n -attribute table T and n -vector of error tolerances \bar{e} ;
Bayesian network G on the set of attributes X and function
neighborhood() defining the "predictive neighborhood" of a
node X_i in G (e.g., $\pi(X_i)$ or $\beta(X_i)$).

Output: A set of materialized (predicted) attributes X_{mat} ($X_{\text{pred}} = X - X_{\text{mat}}$) and a CaRT predictor $\text{PRED}(X_i) \rightarrow X_i$ for each $X_i \in X_{\text{pred}}$.

begin

1. $X_{\text{mat}} := X$, $X_{\text{pred}} := \emptyset$
2. $\text{PRED}(X_i) := \emptyset$ for all $X_i \in X$, improve := true
3. while (improve \neq false) do
4. for each $X_i \in X_{\text{mat}}$
5. mater_neighbors (X_i) :=
 $(X_{\text{mat}} \cap \text{neighborhood}(X_i)) \cup \{\text{PRED}(X) : X \in \text{neighborhood}(X_i), X \in X_{\text{pred}}\} - \{X_i\}$
6. $M := \text{BuildCaRT}(\text{Mater_neighbors}(X_i) \rightarrow X_i, e_i)$
7. let $\text{PRED}(X_i) \subseteq \text{mater_neighbors}(X_i)$ be the set of
predictor attributes used in M
8. cost_change _{i} := 0
9. for each $X_j \in X_{\text{pred}}$ such that $X_i \in \text{PRED}(X_j)$
10. $\text{NEW_PRED}_i(X_j) := \text{PRED}(X_j) - \{X_i\} \cup \text{PRED}(X_i)$
11. $M := \text{BuildCaRT}(\text{NEW_PRED}_i(X_j) \rightarrow X_j, e_j)$
12. set $\text{NEW_PRED}_i(X_j)$ to the (sub)set of
predictor attributes used in M
13. cost_change _{i} := cost_change _{i} + (PredCost($\text{PRED}(X_j) \rightarrow X_j$) - PredCost($\text{NEW_PRED}_i(X_j) \rightarrow X_j$))
14. end
15. end
16. build an undirected, node-weighted graph $G_{\text{temp}} = (X_{\text{mat}}, E_{\text{temp}})$ on the current set of materialized
attributes X_{mat} , where:
17. (a) $E_{\text{temp}} := \{(X, Y) : \text{for all pairs } X, Y \in X_{\text{pred}}\} \cup \{(X_i, Y) : \text{for all } Y \in X_{\text{mat}}\}$
18. (b) weight (X_i) := MaterCost (X_i) - PredCost ($\text{PRED}(X_i) \rightarrow X_i$) + cost_change _{i} for each $X_i \in X_{\text{mat}}$
19. $S := \text{FindWMIS}(G_{\text{temp}})$ /* select (approximate) maximum
weight independent set in G_{temp} as "maximum-benefit" subset of
predicted attributes */
20. if ($\sum_{X \in S} \text{weight}(X) \leq 0$) then improve := false
21. else /* update X_{mat} , X_{pred} , and the chosen CaRT predictors */
22. for each $X_j \in X_{\text{pred}}$
23. if ($\text{PRED}(X_j) \cap S = \{X_i\}$) then $\text{PRED}(X_j) :=$
 $\text{NEW_PRED}_i(X_j)$
24. end
25. $X_{\text{mat}} := X_{\text{mat}} - S$, $X_{\text{pred}} := X_{\text{pred}} \cup S$
26. end
27. end /* while */
28. end

FIGURE 4: The MaxIndependentSet CaRT Selection Algorithm

procedure LowerBound (N, e_i, b)

Input: Leaf N for which lower bound on subtree cost is to be computed; error tolerance e_i for attribute X_i ; bound b on the maximum number of internal nodes in subtree rooted at N .

Output: Lower bound $L(N)$ on cost of subtree rooted at N .

begin

```

1.  for i := 1 to r
2.      minOut [i,0] := i
3.  for J := 1 to b + 1
4.      minOut[0,j] := 0
5.  l := 0
6.  for i := 1 to r
7.      while  $x_i - x_{l+1} > 2_{e_i}$ 
8.          l := l + 1
9.  for j := 1 to b + 1
10.     minOut[i,j] := min {minOut[i - 1,j] + 1, minOut [l,j-1]}
11. end
12. L(N) :=  $\infty$ 
13. for J := 0 to b
14.     L(N) := min {L(N),  $2j + 1 + j \log (|X_i|) + (j + 1 + \text{minOut}(r, j+1)) \log (|\text{dom}(X_i)|)$ }
15. L(N) := min {L(N),  $2b + 3 + (b + 1) \log (|X_i|) + (b + 2) \log (|\text{dom}(X_i)|)$ }
16. return L(N)
end

```

FIGURE 5: Algorithm for Estimating Lower Bound on Subtree Cost

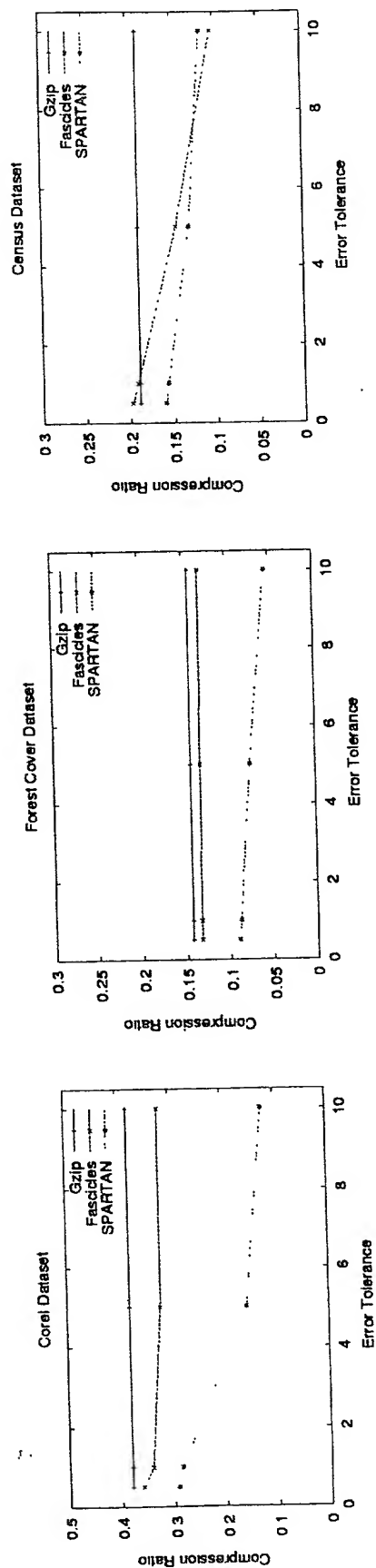


FIGURE 6

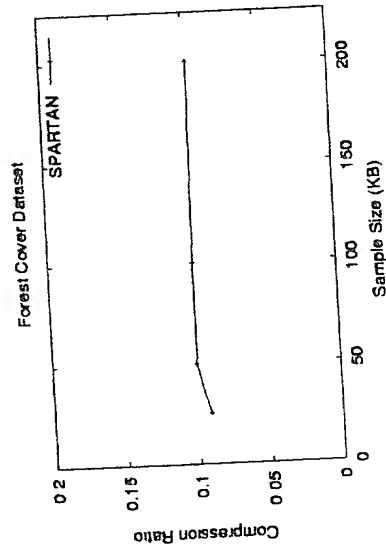
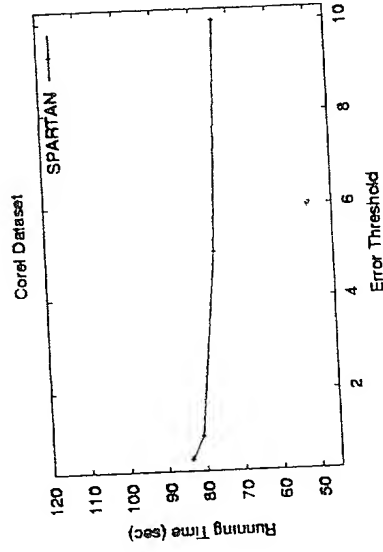
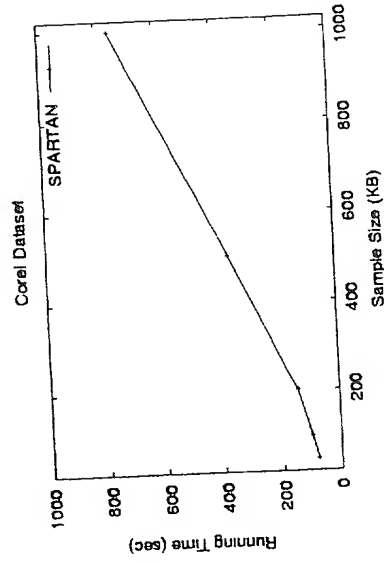


FIGURE 7.